Matrices à blocs et en forme canonique

Guillaume Cano & Maxime Dénès

INRIA Sophia-Antipolis Méditerranée
Guillaume.Cano@inria.fr, Maxime.Denes@inria.fr

Résumé

Nous présentons une formalisation réalisée avec Coq visant essentiellement à prouver l'existence des formes matricielles canoniques de Frobenius et de Jordan, ainsi que leurs propriétés. Nous définissons formellement des notions importantes, comme les matrices diagonales par blocs ou les matrices compagnes, et prouvons des résultats intermédiaires originaux, comme le théorème fondamental de similitude sur un corps, ou encore l'unicité de la forme normale de Smith. Outre la formalisation de la théorie de la réduction des endormorphismes des espaces vectoriels de dimension finie, ce travail ouvre la voie à la certification d'algorithmes efficaces de calcul du polynôme caractéristique ou de la forme de Frobenius.

Introduction

La vérification d'algorithmes de calcul formel est une problématique importante, à la frontière de la vérification de programmes et de la formalisation des mathématiques. Les enjeux sont multiples : réduire le fossé entre une preuve de correction d'algorithme sur papier et un programme implémenté, mais aussi pouvoir bénéficier, au sein d'un assistant de preuve, d'outils habituellement fournis par les logiciels de calcul formel.

Prouver la correction de tels algorithmes requiert le plus souvent d'avoir au préalable formalisé les objets mathématiques mis en jeu et leurs propriétés, avant de se préoccuper des détails d'implémentation et des aspects de performance. C'est dans cet esprit que nous prouvons l'existence et les propriétés de la forme matricielle canonique de Frobenius, puis en déduisons celles de la forme de Jordan.

Ces formes jouent un rôle important en algèbre linéaire car elles caractérisent la similitude de matrices, respectivement sur un corps pour la forme de Frobenius et sur un corps algébriquement clos pour celle de Jordan. Ainsi, étudier le rang, la trace, le déterminant, les valeurs propres, le polynôme minimal ou caractéristique d'une matrice revient à l'étudier sur sa forme normale, ce qui peut simplifier le problème.

À titre d'exemple, la forme de Frobenius donne un moyen efficace d'élever une matrice à de grandes puissances [Gie95]. Ceci peut être utile pour calculer l'état d'une chaîne de Markov à l'instant n, où n est grand. En effet, ce calcul revient à élever la matrice de transition de la chaîne à la puissance n.

Dans cet article, nous décrivons un développement formel que nous avons réalisé à l'aide de l'assistant à la preuve Coq [Coq12] et de la bibliothèque SSReflect [GM09], initialement développée pour la preuve du théorème des quatre couleurs puis dans le projet *Mathematical Components* ayant récemment abouti à la preuve du théorème de Feit-Thompson.

Dans la section 1, nous présentons notre définition de matrices diagonales par blocs, et en particulier comment nous traitons les problèmes liés au fait que les matrices dans SSReflect sont dépendamment typées. La section 2 revient sur les notions d'équivalence et de similitude de matrices, ainsi que les liens entre elles, et leur formalisation. Dans la section 3, nous réutilisons une formalisation existante de la forme normale de Smith [CDMS12] et en prouvons l'unicité, pour obtenir la théorie des facteurs

invariants. Ensuite, nous en déduisons l'existence de la forme de Frobenius (section 4), puis de celle de Jordan (section 5) ainsi que les propriétés habituelles de diagonalisation des endomorphismes d'un espace vectoriel de dimension finie.

1. Matrices diagonales par blocs

Le type des matrices de taille $m \times n$ sur un anneau R est noté `'M[R]_(m,n)` dans SSReflect (`'M[R]_m` pour des matrices carrées). En particulier, les matrices ont un type dépendant, indexé par leur taille. Ce choix de conception a plusieurs conséquences importantes. Parmi les avantages, il rend plus concis la plupart des énoncés de lemmes traitant de matrices. En effet, si la taille n'apparaissait pas dans le type, il serait souvent nécessaire de rajouter des conditions explicites en prémisse de ces lemmes. Cette information fournie par typage rend également plus aisée l'abstraction de structures algébriques sur les matrices, par exemple pour inférer une structure d'anneau sur les matrices carrées.

Mais ce choix a aussi des inconvénients, notamment le fait que le typage considère les tailles seulement modulo convertibilité. Ainsi, `'M[R]_(m,n)` et `'M[R]_(0 + m, n)` sont bien un seul et même type, mais `'M[R]_(m + 0, n)` en est un autre. Une contrainte rend ce problème particulièrement visible : la structure d'anneau telle que définie dans SSReflect exclut l'anneau trivial. En particulier, les matrices carrées ne forment un anneau que lorsque leur taille est convertible au successeur d'un entier (i.e. à un terme de la forme `m.+1`).

Deux fonctions définies dans la bibliothèque permettent de manipuler le type des matrices :

- La fonction `castmx` qui, étant donné une matrice de type `'M[R]_(m1,n1)` et une preuve que `m1 = m2` et `n1 = n2`, rend une matrice de type `'M[R]_(m2,n2)` ayant les mêmes coefficients.
- La fonction `conform_mx` qui prend en arguments deux matrices `A` et `B` de types respectifs `'M[R]_(m1,n1)` et `'M[R]_(m2,n2)`, et renvoie `B` si `m1 = m2` et `n1 = n2`, `A` sinon.

La notion de matrice diagonale par blocs n'a de sens que si on spécifie le découpage des blocs. L'intérêt de telles matrices est que de nombreuses opérations, comme l'addition, peuvent s'exprimer bloc par bloc, à condition toutefois que les deux matrices auxquelles on applique l'opération aient le même découpage en blocs.

Une première idée serait de représenter une matrice diagonale par blocs par une séquence de paires dépendantes (où chaque élément de la séquence est un bloc indexé par sa taille). Cependant, cela ne permettrait pas d'exprimer par le typage que deux matrices ont des découpages compatibles.

Notre définition de matrice diagonale par blocs prend donc en premier argument une liste d'entiers naturels qui expriment la dimension de chaque bloc. Nous décidons de représenter les blocs eux-mêmes par une fonction `F : forall (n : nat), nat -> 'M[R]_n`. Ainsi, si le bloc numéro `i` a pour taille `n`, il sera représenté par `F n i`, ou encore, si la liste des tailles des blocs est `s`, par `F s`_i i`.

Pour construire une matrices à partir de ses blocs, nous utilisons la fonction `block_mx`. Plus précisément, si `A`, `B`, `C` et `D` sont des matrices de dimensions compatibles, `block_mx A B C D` représente la matrice :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

Un premier essai pour décrire les matrices diagonales par blocs consiste à appliquer récursivement `block_mx` :

```
Fixpoint diag_block_mx (s : seq nat) (F : forall (n : nat), nat -> 'M[R]_n) :=
  if s is n :: s' return 'M_(sumn s)
   then block_mx (F n 0) 0 0 (diag_block_mx s' (fun n i => F n i.+1))
   else 0.
```

où `sumn s` est la somme des éléments de la séquence d'entiers `s`.

Mais cette définition ne remplit pas la condition que nous évoquions plus haut, d'avoir une taille convertible à un successeur. Dès lors, il sera impossible d'appliquer les opérations d'anneau aux matrices diagonales par blocs, ce qui était pourtant une des motivations initiales pour les définir.

Nous devons donc trouver un moyen de mettre en évidence un successeur dans la taille des matrices diagonales par blocs. Pour que ce soit possible, il faut exiger qu'il y ait au moins un bloc, et qu'un des blocs soit non-vide. De plus, si on veut pouvoir appliquer les opérations d'anneau bloc par bloc, il faut même imposer que tous les blocs soient non-vides. Le type de `F` devient donc `forall (n : nat), nat -> 'M[R]_n.+1`, et on définit les matrices diagonales par blocs en deux étapes, d'abord les fonctions auxiliaires suivantes :

```
Fixpoint size_sum_rec k (s : seq nat) : nat :=
  if s is x :: l then k + (size_sum_rec x l).+1 else k.

Fixpoint diag_block_mx_rec k (s : seq nat)
 (F : (forall n, nat -> 'M[R]_n.+1)) :=
  if s is x :: l return 'M_((size_sum_rec k s).+1)
   then block_mx (F k 0) 0 0 (diag_block_mx_rec x l (fun n i => F n i.+1))
   else F k 0.
```

Puis les fonctions principales :

```
Definition size_sum s := if s is x :: l then size_sum_rec x l else 0.

Definition diag_block_mx s F :=
  if s is x :: l return 'M_((size_sum s).+1)
  then diag_block_mx_rec x l F else 0.
```

Nous avons défini la fonction `size_sum` de telle manière que la matrice construite par `diag_block_mx s F` soit de taille `(size_sum s).+1`. Avec cette nouvelle définition, la séquence `s` décrit les prédécesseurs des tailles des blocs (et non directement les tailles).

Notre définition de matrices diagonales par blocs est donc rendue moins naturelle du fait de cette contrainte sur la taille. Autoriser l'anneau trivial dans la définition des anneaux de SSReflect semble possible, mais serait moins confortable pour le traitement de certaines théories sur ces anneaux, notamment le développement des polynômes à une variable, qui y sont définis comme des listes de coefficients dont le dernier est non nul (l'ensemble des polynômes sur l'anneau trivial serait alors vide, en particulier ce ne serait pas un anneau).

2. Equivalence et similitude de matrices

L'équivalence et la similitude sont des notions fondamentales en algèbre matricielle. Deux matrices A et B (non nécessairement carrées) sont équivalentes s'il existe deux matrices inversibles M et N telles que $MAN = B$. Ceci exprime que les systèmes linéaires représentés par A et B admettent le même espace de solutions (à isomorphisme près).

Par ailleurs, deux matrices carrées A et B sont semblables s'il existe une matrice inversible P telle que $PAP^{-1} = B$ (ou encore $PA = BP$). La similitude est une propriété plus forte que l'équivalence, mais ne s'applique qu'à des matrices carrées, et exprime un changement de base : si A représente un endomorphisme f dans une base $\mathcal{B}$ et est semblable à B, alors B représente également f dans une base $\mathcal{B}'$.

Nous définissons formellement la similitude de matrices comme suit :

```
Definition similar m n (A : 'M[R]_m) (B : 'M[R]_n) := m = n /\
  (exists P : 'M_m , P \in unitmx /\ P *m A = (conform_mx P B) *m P).
```

où `unitmx` est un prédicat désignant l'inversibilité et `*m` est une notation pour la multiplication des matrices.

Cette définition relâche le type des matrices en arguments : `similar` peut être appliqué à des matrices `A` et `B` ayant des tailles non convertibles, mais `similar A B` ne sera prouvable que si `A` et `B` ont des tailles prouvablement égales. Nous utilisons la même astuce pour la définition de l'équivalence :

```
Definition equivalent m1 n1 m2 n2 (A : 'M[R]_(m1,n1)) (B : 'M[R]_(m2,n2)) :=
  [/\ m1 = m2, n1 = n2 & exists M, exists N,
  [/\ M \in unitmx , N \in unitmx & M *m A *m N = conform_mx A B]].
```

Un lien important entre ces deux notions est le fait que deux matrices carrées sont semblables si et seulement si leurs matrices caractéristiques sont équivalentes :

```
Theorem similar_fundamental m n (A: 'M[R]_m) (B : 'M[R]_n) :
  similar A B <-> equivalent (char_poly_mx A) (char_poly_mx B).
```

Ici `char_poly_mx A` désigne la matrice caractéristique de `A`, il s'agit de la matrice $XI - A$ où X est l'indéterminée de l'anneau de polynômes $R[X]$.

Ce résultat est parfois appelé « théorème fondamental de similitude sur un corps ». Pour l'établir, nous suivons la preuve décrite dans [Wed08].

Une des deux implications est facile : si A et B sont semblables, il existe une matrice inversible P telle que $A = PBP^{-1}$ et donc $XI - A = P(XI - B)P^{-1}$, ce qui implique que $XI - A$ et $XI - B$ sont équivalentes.

La réciproque est plus difficile. Supposons qu'il existe des matrices inversibles M et N telles que :

$$M(XI - A)N = XI - B$$

Jusqu'à présent, les objets que nous manipulions étaient des matrices de polynômes. Mais nous devons à présent les voir formellement comme des polynômes de matrices, ce qui est possible en utilisant l'isomorphisme suivant :

$$\phi : M(R[X]) \to M(R)[X]$$

Cet isomorphisme était déjà un des ingrédients clé dans la preuve formelle du théorème de Cayley-Hamilton décrit dans [OB08]. Il a été défini dans la bibliothèque SSReflect de la manière suivante :

```
Definition phi n (A : 'M[{poly R}]_n.+1) :=
  \poly_(k < \max_i \max_j size (A i j)) \matrix_(i, j) (A i j)`_k.
```

Les notations `\poly_(i < n) a i` et `\matrix_(i < m, j < n) M i j` permettent de définir respectivement un polynôme et une matrice par l'expression générale de leurs coefficients. Dans la bibliothèque SSReflect un polynôme `p` est vu comme une séquence de coefficients. De ce fait, `size p` désigne la taille de cette séquence et, si `p` est non nul, son degré est représenté par `(size p).-1`.

Nous définissons ensuite les polynômes de matrices M_1, M_0 et N_1, N_0 respectivement par division à gauche de $\phi(M)$ et division à droite de $\phi(N)$:

$$\phi(M) = (X - B)M_1 + M_0 \quad \text{avec} \quad \deg M_0 = 0$$

$$\phi(N) = N_1(X - B) + N_0 \quad \text{avec} \quad \deg N_0 = 0$$

L'étape clé de cette preuve consiste à établir l'identité suivante :

$$M_0(X - A)N_0 = (1 - (X - B)R_1)(X - B)$$

avec $R_1 = M_1 * \phi(M^{-1}) + \phi(N^{-1}) * N_1 - M_1 * (X - A) * N_1$

Ceci se fait par des manipulations algébriques élémentaires. Ensuite, puisque le degré du membre gauche est 1 (M_0 et N_0 sont des constantes), R_1 doit être nul (si $R_1 \neq 0$, le membre droit aurait un degré au moins 2). L'identité précédente devient alors :

$$M_0(X - A)N_0 = (X - B)$$

À partir de quoi on peut déduire, en identifiant les coefficients :

$$M_0 N_0 = 1$$

$$M_0 A N_0 = B$$

D'où $M_0 = N_0^{-1}$, $N_0^{-1} A N_0 = B$, c'est-à-dire que A et B sont semblables.

Même si cette preuve peut paraître naturelle, les objets d'étude (des polynômes sur un anneau non commutatif et non intègre [1] en général) demandent des précautions, car il ne bénéficient évidemment pas de toutes les propriétés habituelles des polynômes sur un anneau commutatif intègre. L'utilisation d'un assistant de preuve permet d'éviter la tentation d'arguments par analogie incorrects.

Raisonner sur le degré de tels polynômes, par exemple, requiert des lemmes spécifiques pour les polynômes unitaires, ou plus généralement, pour le cas où le coefficient dominant est un élément régulier de l'anneau de base. Heureusement, la bibliothèque SSReflect fournit les niveaux d'abstraction adéquats.

3. Forme normale de Smith

On dit qu'une matrice est en forme normale de Smith si elle se présente de la manière suivante :

$$\begin{pmatrix} d_1 & 0 & \dots & \dots & & \dots & 0 \\ 0 & d_2 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \dots & 0 & d_k & 0 & \dots & 0 \end{pmatrix}$$

avec la particularité que $\forall i, 1 \leq i < k, d_i \mid d_{i+1}$.

Toute matrice sur un anneau principal est équivalente à une matrice en forme normale de Smith. Ce résultat peut être vu comme une généralisation de l'élimination gaussienne pour une matrice sur un corps (qui permet d'arriver à une forme diagonale avec seulement des 1, et était déjà connue des Chinois au IIème siècle [CSLY04]).

Nous utilisons dans cette section une formalisation déjà existante d'un algorithme de mise en forme normale de Smith [CDMS12] pour assurer son existence, et nous établissons, indépendamment de toute implémentation particulière, l'unicité de cette forme et le fait qu'elle caractérise donc l'équivalence de deux matrices sur un anneau principal. Enfin, en appliquant ces outils aux matrices caractéristiques, nous obtenons les facteurs invariants d'une matrice sur un corps, ainsi que leurs propriétés.

3.1. Existence

Pour représenter formellement la forme normale de Smith, nous utilisons donc la fonction `diag_mx_seq` qui représente une matrice diagonale dont les éléments diagonaux sont donnés par une séquence passée en argument. Elle est définie comme suit :

```
Definition diag_mx_seq m n s := \matrix_(i < m, j < n) s`_i *+ (i == j :> nat).
```

1. Un anneau est intègre si tous ses éléments non nuls sont réguliers, i.e. non diviseurs de zéro.

La notation `x *+ n`, où `x` est élément d'un anneau et `n` un entier naturel, désigne la somme `x +...+ x` itérée `n` fois. Dans l'expression du coefficient général de la matrice ci-dessus, `i` et `j` sont des ordinaux (i.e. des entiers bornés respectivement par `m` et `n`), ils ont donc des types distincts (notés respectivement `'I_m` et `'I_n`). La notation `i == j :> nat` permet d'indiquer à Coq de les comparer en tant qu'entiers naturels et de renvoyer un booléen. Une coercion envoie ensuite ce booléen sur un entier (`true` est interprété par `1` et `false` par `0`). Ainsi, `s`_i *+ (i == j :> nat)` désigne l'élément d'indice `i` dans `s` si `i` et `j` ont la même valeur, `0` sinon.

Nous disposons d'une fonction `Smith` qui, appliquée à une matrice `A`, retourne une liste `s` et des matrices `L0` et `R0` telles que :

- La séquence `s` est triée pour la relation de division.
- La matrice `diag_mx_seq m n s` est équivalente à `A`, avec pour matrices de passage `L0` et `R0`

Ce qui se traduit formellement par un prédicat inductif :

```
CoInductive Smith_spec {m n} M : 'M[R]_m * seq R * 'M[R]_n -> Type :=
  SmithSpec L0 d R0 of L0 *m M *m R0 = diag_mx_seq m n d
    & sorted (@dvdr R) d
    & L0 \in unitmx & R0 \in unitmx : Smith_spec M (L0, d, R0).
```

Et le lemme de correction suivant :

```
Lemma SmithP : forall (m n : nat) (M : 'M_(m,n)), Smith_spec M (Smith M).
```

3.2. Unicité

L'algorithme de mise en forme de Smith met en oeuvre des calculs de pgcd, dont le résultat n'est unique que modulo association (c'est-à-dire la relation $\sim$ définie par $a \sim b$ ssi $a|b$ et $b|a$). On ne peut donc pas donc espérer associer à toute matrice une forme de Smith strictement unique, mais nous prouvons l'unicité modulo la relation $\sim$ sur les coefficients diagonaux. Comme nous sommes dans le cas d'un anneau intègre, cela revient à dire que la forme de Smith est unique modulo la multiplication des coefficients diagonaux par des inversibles.

Pour établir ce résultat d'unicité, nous prouvons que les coefficients de la forme de Smith d'une matrice quelconque A s'expriment à partir de pgcd de mineurs de A (i.e. le déterminant des sous-matrices de A). Plus précisément, si on note $\wedge$ le pgcd et $|A|_k$ l'ensemble des mineurs d'ordre k de A, on a l'identité suivante :

$$\prod_{i=1}^{k} d_i \sim \bigwedge_{x \in |A|_k} x$$

Pour exprimer ce résultat formellement, nous utilisons des notions de sous-matrices et de mineurs définis, suivant [Sil12], à l'aide de fonctions de ré-indexation :

```
Variables p q m n : nat.

Definition submatrix (f : 'I_p -> 'I_m) (g : 'I_q -> 'I_n) (A : 'M_(m,n)) :=
  \matrix_(i < p, j < q) A (f i) (g j).

Definition minor (f : 'I_p -> 'I_m) (g : 'I_p -> 'I_n) (A : 'M_(m,n)) :=
  \det (submatrix f g A).
```

L'identité que nous cherchons à prouver s'exprime maintenant, pour une séquence `s` et une matrice `A` satisfaisant la spécification `Smith_spec`, comme suit :

```
Lemma Smith_gcdr_spec :
  \prod_(i < k) s`_i %= \big[gcdr/0]_f \big[gcdr/0]_g minor k f g A .
```

où `%=` dénote la relation d'association $\sim$. La notation avec `\big` indique ici que l'on itère le pgcd.

Dans une première étape nous prouvons le théorème pour la matrice `diag_mx_seq n n s` (au lieu de `A`). Comme c'est une matrice diagonale, les seuls mineurs d'ordre k non nuls sont les produits de k éléments de la séquence `s`. Et comme chaque élément de la séquence divise le suivant, le pgcd des mineurs d'ordre k est donc le produit des k premiers éléments de la séquence `s`.

Dans la seconde étape, il nous reste à prouver que le pgcd des mineurs d'ordre k de la matrice `A` est le même que le pcgd de ceux de `diag_mx_seq m n s` :

```
\big[gcdr/0]_f \big[gcdr/0]_g minor k f g (diag_mx_seq m n s)
  %= \big[gcdr/0]_f \big[gcdr/0]_g minor k f g A
```

Il faut donc montrer que les deux membres de l'égalité se divisent l'un l'autre. Comme ces deux preuves se font exactement de la même manière, nous montrons ici seulement que le membre droit divise le membre gauche.

Pour diviser un pgcd, il suffit de diviser tous les éléments dont on calcule le pgcd. Il nous reste donc à montrer que pour tout `f` et `g`, le membre droit divise `minor k f g (diag_mx_seq m n s)`. D'autre part, la matrice `diag_mx_seq m n s` est équivalente à `A`, donc il existe deux matrices `M` et `N` telles que `diag_mx_seq m n s = M *m A *m N`. Il faut donc montrer que le membre droit divise `minor k f g (M *m A *m N)`, qui peut s'exprimer comme le déterminant d'un produit. À ce stade, nous utilisons la preuve formelle de la formule de Binet-Cauchy [Sil12] qui s'énonce ainsi :

$$\det(AB) = \sum_{\substack{I \in \mathcal{P}(\{1,\dots,l\}) \\ \#|I|=k}} \det(A_I)\det(B_I)$$

où A est une matrice de taille $k \times l$ et B une matrice de taille $l \times k$. A_I (resp. B_I) est la matrice constituée des k colonnes (resp. lignes) de A (resp. B) dont les indices sont dans I.

Ce théorème nous permet de transformer l'expression `minor k f g (M *m A *m N)` en une somme de mineurs. Or pour diviser une somme, il suffit de diviser chacun de ses termes, ce qui nous amène à devoir prouver que pour tout `h` et `i`, on a :

```
\big[gcdr/0]_f \big[gcdr/0]_g minor k f g A %| minor k h i A
```

Ce qui est vrai par définition du pgcd.

Le lemme précédent utilisé avec $k = 1$ permet de déterminer de manière unique (modulo la relation d'association $\sim$) le premier élément diagonal de la forme de Smith. Puis il détermine ainsi, de proche en proche, tous les éléments diagonaux. Parmi les matrices équivalentes modulo $\sim$ à la forme de Smith d'une matrice `A`, nous avons fixé un représentant de même déterminant que `A` :

```
Definition Smith_form m n (A : 'M[R]_(m,n)) := diag_mx_seq m n (Smith_seq A).

Lemma det_Smith n (A :'M[R]_n) : \det (Smith_form A) = \det A.
```

3.3. Facteurs invariants

Soit F un corps, et A une matrice à coefficients dans F. Nous allons appliquer l'algorithme de Smith à la matrice caractéristique $XI - A$ de A. D'après le lemme `det_Smith` évoqué plus haut, le déterminant de la forme normale de Smith de $XI - A$ est le polynôme caractéristique de A. Ce qui nous assure qu'aucun élément diagonal de la forme normale de Smith n'est nul pour les deux raisons suivantes :

- Le déterminant de la forme normale de Smith est le produit de ses coefficients diagonaux.
- Le polynôme caractéristique d'une matrice n'est jamais nul.

Les coefficients diagonaux sont donc des polynômes non nuls à coefficients dans un corps, on peut donc diviser chacun de ces polynômes par son coefficient dominant pour avoir des polynômes unitaires :

```
Definition Frobenius_seq n (A : 'M[F]_n) :=
  [seq (lead_coef p)^-1 *: p | p <- (take n (Smith_seq (char_poly_mx A)))].
```

où `take n s` est la séquence des `n` premiers éléments de la séquence `s`. L'algorithme de Smith nous renvoie une séquence, mais ne nous donne aucune information sur sa taille. Ici l'utilisation de la fonction `take` permet de montrer que `size (Frobenius_seq A) = n`. C'est un résultat qui sera utile dans le développement formel. L'utilisation de la fonction `take` ne change rien pour la forme normale de Smith comme le montre le résultat suivant :

```
Lemma diag_mx_seq_take n s : diagmx_seq n n s = diag_mx_seq n n (take n s).
```

Les facteurs invariants sont les polynômes non constants de `Frobenius_seq` :

```
Definition invariant_factors n (A : 'M[F]_n) :=
  [seq p : {poly R} <- (Frobenius_seq A) | 1 < size p].
```

Nous avons défini les facteurs invariants de telle manière qu'ils soient unitaires, car plus loin nous parlerons des matrices compagnes de ces polynômes, or les propriétés usuelles des matrices compagnes ne sont vraies que pour les polynômes unitaires.

4. Forme de Frobenius

La forme normale de Frobenius d'une matrice M est une matrice diagonale par blocs dont les blocs sont les matrices compagnes des facteurs invariants de M. Nous montrons donc d'abord notre formalisation des matrices compagnes avant de donner une définition formelle de la forme normale de Frobenius. Nous présentons ensuite un schéma de la preuve formelle qu'une matrice et sa forme normale de Frobenius sont semblables.

4.1. Matrices compagnes

La matrice compagne d'un polynôme $p = X^n + a_{n-1}X^{n-1} + \ldots + a_1X + a_0$ est la matrice suivante :

$$\mathcal{C}_p = \begin{pmatrix} 0 & \ldots & 0 & 0 & -a_0 \\ 1 & \ddots & \vdots & \vdots & -a_1 \\ 0 & \ddots & 0 & \vdots & \vdots \\ \vdots & \ddots & 1 & 0 & \vdots \\ 0 & \ldots & 0 & 1 & -a_{n-1} \end{pmatrix}$$

Cette matrice est intéressante car p est à la fois son polynôme caractéristique et son polynôme minimal.

Formellement, si `p` est un polynôme non constant, la dimension de la matrice compagne de `p` est `(size p).-1`. Mais ici encore, pour les raisons vues précédemment, la taille d'une matrice compagne doit être convertible à un successeur. Nous définissons donc les matrices compagnes de sorte que leur taille soit `(size p).-2.+1`, qui sera prouvablement égal (bien que non convertible) à `(size p).-1`, pour peu que nous nous restreignions aux polynômes non constants :

```
Definition companion_mx (p : {poly R}) :=
  \matrix_(i,j < (size p).-2.+1)
      ((i == j.+1 :> nat)%:R + p`_i *+ ((size p).-2 == j)).
```

où pour tout entier `n`, `n%:R` est une notation pour `1 *+ n` (c'est-à-dire `1 + ... + 1` avec `n` termes). Cette définition est parfaitement valide, mais elle ne nous permet pas de définir une matrice diagonale par blocs dont les blocs sont des matrices compagnes (ce dont nous aurons besoin pour définir la forme normale de Frobenius).

En effet, le type de `diag_block_mx` impose à la fonction décrivant les blocs d'avoir le type `forall (n : nat), nat -> 'M_n.+1`, mais la définition `companion_mx` plus haut est de type `forall (p : {poly R}), 'M_((size p).-2.+1)`

Pour résoudre ce problème, nous introduisons une définition intermédiaire `companion_mxn` qui relâche la taille de la matrice retournée :

```
Definition companion_mxn n (p : {poly R}) :=
  \matrix_(i,j < n) ((i == j.+1 :>nat )%:R + p`_i *+ ((size p).-2 == j)).

Definition companion_mx (p : {poly R}) := companion_mxn (size p).-2.+1 p.
```

Ainsi, la matrice `diag_block_mx s companion_mxn` sera bien typée, et aura les propriétés attendues si `s` contient des tailles de la forme `(size p).-2`. Les lemmes sur les matrices compagnes seront exprimés, eux, sur `companion_mx`.

La forme normale de Frobenius d'une matrice A (à coefficients dans un corps) est la matrice suivante :

$$\begin{pmatrix} \mathcal{C}_{p_1} & & & 0 \\ & \mathcal{C}_{p_2} & & \\ & & \ddots & \\ 0 & & & \mathcal{C}_{p_k} \end{pmatrix}$$

où les polynômes p_i sont les facteurs invariants de la matrice A. Formellement, elle peut être définie de la manière suivante :

```
Definition Frobenius_form n (A : 'M[R]_n) :=
  let sizes := [seq (size p).-2 | p : {poly R} <- (invariant_factors A)] in
  let blocks n i := companion_mxn n.+1 (nth 0 (invariant_factors A) i) in
   diag_block_mx sizes blocks.
```

4.2. De Smith à Frobenius

Nous allons maintenant montrer que toute matrice sur un corps `F` est semblable à sa forme de Frobenius :

```
Lemma Frobenius n (A : 'M[F]_n.+1) : similar A (Frobenius_form A).
```

Le théorème `similar_fundamental` décrit section 2 montre que pour prouver ce résultat, il nous suffit d'établir que pour toute matrice `A`, les matrices caractéristiques de `A` et `Frobenius_form A` sont équivalentes.

Nous allons donc partir de la matrice $XI - A$ et, par transitivité de l'équivalence, arriver à la matrice caractéristique de `Frobenius_form A`.

Nous savons que $XI - A$ est équivalente à sa forme normale de Smith qui, en prenant les polynômes diagonaux unitaires, est de la forme :

$$\begin{pmatrix} 1 & & & & & \\ & \ddots & & & 0 & \\ & & 1 & & & \\ & & & p_1 & & \\ & 0 & & & \ddots & \\ & & & & & p_n \end{pmatrix}$$

où les p_i sont les facteurs invariants de la matrice A.

Permuter les éléments diagonaux de la matrice ne change pas le fait que la matrice ainsi obtenue est équivalente à la matrice $XI - A$. Car les matrices de passage de l'une à l'autre sont des matrices de permutations et donc inversibles. L'énoncé formel de ce résultat se présente de la manière suivante :

```
Lemma similar_diag_mx_seq m n s1 s2 :
  m = n -> size s1 = m -> perm_eq s1 s2 ->
  similar (diag_mx_seq m m s1) (diag_mx_seq n n s2).
```

où `perm_eq s1 s2` exprime le fait que les séquence `s1` et `s2` ont les mêmes éléments mais permutés. Nous pouvons ainsi permuter les éléments diagonaux de la matrice précédente :

$$\begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & 0 & \\ & & 1 & & & & & \\ & & & p_1 & & & & \\ & & & & \ddots & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & 0 & & & & & & 1 \\ & & & & & & & & p_n \end{pmatrix}$$

Si `pi` représente le polynôme p_i, alors le nombre de 1 avant p_i est `(size pi).-2`. Cette matrice peut être vue comme la matrice diagonale par blocs suivante :

$$\left(\begin{array}{ccc|ccc} 1 & & & & & \\ & \ddots & & & 0 & \\ & & 1 & & & \\ & & & p_1 & & \\ \hline & & & \ddots & & \\ & & & & 1 & \\ & 0 & & & & \ddots \\ & & & & & & 1 \\ & & & & & & & p_n \end{array}\right)$$

Il nous reste maintenant à montrer que cette matrice est équivalente à la matrice caractéristique de la forme normale de Frobenius. Nous utilisons d'abord le fait que la matrice caractéristique d'une matrice diagonale par blocs est la matrice diagonale par blocs des blocs caractéristiques :

```
Lemma char_diag_block_mx s (F : forall n, nat -> 'M[R]_n.+1) :
  s != [::] ->
  char_poly_mx (diag_block_mx s F) =
  diag_block_mx s (fun n i => char_poly_mx (F n i)).
```

Il ne nous reste donc plus qu'à montrer que la matrice précédente est équivalente à :

$$\begin{pmatrix} XI - \mathcal{C}_{p_1} & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & XI - \mathcal{C}_{p_n} \end{pmatrix}$$

Pour cela, il suffit de prouver que les matrices sont équivalentes bloc à bloc, c'est-à-dire que pour chaque indice i, la matrice $XI - \mathcal{C}_{p_i}$ est équivalente à :

$$\begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & p_i \end{pmatrix}$$

Or il se trouve que cette matrice est la forme normale de Smith de la matrice $XI - \mathcal{C}_{p_i}$. Pour montrer ce dernier résultat, nous utilisons le lemme `Smith_gcdr_spec` mentionné dans la section 3. En effet, pour tout `k` tel que `k < (size pi).-2`, on peut trouver une sous-matrice de la matrice $XI - \mathcal{C}_{p_i}$ ci-dessous n'ayant que des -1 sur la diagonale :

$$XI - \mathcal{C}_{p_i} = \begin{pmatrix} X & \dots & 0 & 0 & a_0 \\ -1 & \ddots & \vdots & \vdots & a_1 \\ 0 & \ddots & X & \vdots & \vdots \\ \vdots & \ddots & -1 & X & \vdots \\ 0 & \dots & 0 & -1 & X + a_{n-1} \end{pmatrix}$$

C'est-à-dire que pour tout tel `k`, cette matrice a un mineur d'ordre `k` associé à 1 (car $(-1)^k \sim 1$), et donc le pcgd de ces mineurs est lui-même associé à 1. Il est donc possible choisir les `(size pi).-2` premiers éléments diagonaux de la forme normale de Smith de la matrice $XI - \mathcal{C}_{p_i}$ tels qu'ils soient égaux à 1. Pour le dernier élément diagonal, le seul choix possible est le polynôme p_i, car le produit des éléments diagonaux est le déterminant de la forme normale de Smith, et est également le déterminant de la matrice $XI - \mathcal{C}_{p_i}$.

5. Forme de Jordan, trigonalisation et diagonalisation

La forme normale de Jordan d'une matrice A est une matrice triangulaire supérieure dont les éléments diagonaux sont les racines du polynôme caractéristique de la matrice A (c'est-à-dire les valeurs propres de A). Pour que cette forme existe, il suffit que le polynôme caractéristique soit scindé à racines simples. Afin d'assurer cette condition, nous choisissons de travailler sur un corps algébriquement clos `F`.

Nous récapitulons dans un premier temps les éléments de la théorie des corps algébriquement clos [Coh12] que nous utilisons. Nous définissons ensuite la forme normale de Jordan, puis nous montrons comment l'obtenir à partir de celle de Frobenius.

5.1. Polynômes à coefficients dans un corps alébriquement clos

Si p est un polynôme à coefficients dans un corps clos tel que

$$p = \prod_{i=1}^{m} (X - \lambda_i)^{\mu_i}$$

alors :

- `root_seq p` est la séquence des λ_i.
- `root_mu_seq p` est la séquence des paires (μ_i, λ_i).
- `linear_factor_seq p` est la séquence des polynômes $(X - \lambda_i)^{\mu_i}$.
- Si `s` est une séquence de polynômes, alors `root_seq_poly s` est la concaténation de toutes les séquences obtenues lorsque l'on applique `root_mu_seq` sur chacun des polynômes de `s`.

5.2. Définitions

On appelle bloc de Jordan la matrice suivante :

$$J(\lambda, n) = \begin{pmatrix} \lambda & 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \dots & \dots & 0 & \lambda \end{pmatrix}$$

Nous le définissons formellement de la manière suivante :

```
Definition Jordan_block lam n : 'M[F]_n :=
  \matrix_(i,j) (lam *+ (i == j) + (i.+1 == j)%:R).
```

La forme normale de Jordan est une matrice diagonale par blocs composée de blocs de Jordan :

```
Definition Jordan_form n (A : 'M[R]_n.+1) :=
  let sp := root_seq_poly (invariant_factors A) in
  let sizes := [seq x.1 | x <- sp] in
  let blocks n i := Jordan_block (nth (0,0) sp i).2 n.+1 in
  diag_block_mx sizes blocks.
```

Dans la suite nous expliquons le passage de la forme normale de Frobenius à celle de Jordan. Cela permettra de mieux comprendre la définition donnée ci-dessus.

5.3. De Frobenius à Jordan

Soit A une matrice à coefficients dans un corps clos `F`. Nous avons déjà prouvé que la matrice A est semblable à sa forme normale de Frobenius. Nous allons montrer que la forme normale de Frobenius de A est semblable à sa forme normale de Jordan. Par transitivité de la similitude, nous aurons ainsi prouvé que toute matrice est semblable à sa forme normale de Jordan :

```
Lemma Jordan n (A : 'M[F]_n.+1) : similar A (Jordan_form A).
```

On peut sans perte de généralité travailler sur un seul bloc de la forme normale de Frobenius. Fixons donc un indice i, et travaillons sur la matrice $\mathcal{C}_{p_i}$ où p_i est le i-ème facteur invariant de A.

Montrons d'abord que si $q = q_1 \dots q_m$ et que les polynômes q_i sont premiers entre eux deux à deux, alors la matrice $\mathcal{C}_q$ est semblable à :

$$\begin{pmatrix} \mathcal{C}_{q_1} & & 0 \\ & \ddots & \\ & & \ddots \\ 0 & & & \mathcal{C}_{q_m} \end{pmatrix}$$

Par récurrence sur m il suffit de montrer que $\mathcal{C}_q$ est semblable à :

$$\begin{pmatrix} \mathcal{C}_{q_1} & 0 \\ 0 & \mathcal{C}_{q_2 \dots q_m} \end{pmatrix}$$

Notre première tentative pour prouver l'équivalence ci-dessus a été de donner explicitement la matrice de passage. Cela aboutit, mais c'est une preuve assez longue qui finalement consiste à refaire la preuve de l'algorithme de Smith. Nous avons finalement opté pour une preuve utilisant le théorème `similar_fundamental` de la section 2. Cela nous ramène à prouver une équivalence entre des matrices ne contenant que des formes normales de Smith de matrices compagnes. Pour montrer cette équivalence nous utilisons le lemme `Smith_gcdr_spec` de la même manière que dans la fin de la section 4. Nous en déduisons que $q_1 * q_2 \dots q_m = q$ est le seul facteur invariant de la matrice ci-dessus ainsi que de la matrice $\mathcal{C}_q$. Les deux matrices ont les mêmes facteurs invariants, donc sont équivalentes. Ce qui conclut la démonstration de ce résultat.

Le corps `F` étant clos, nous pouvons décomposer le facteur invariant p_i comme suit :

$$p_i = \prod_{j=1}^{m_i} (X - \lambda_{ij})^{\mu_{ij}}$$

où les λ_{ij} sont les racines de p_i et les μ_i leur multiplicité. Le résultat précédent nous permet d'établir que la matrice compagne $\mathcal{C}_{p_i}$ est semblable à :

$$\begin{pmatrix} \mathcal{C}_{(X-\lambda_{i1})^{\mu_{i1}}} & & 0 \\ & \ddots & \\ & & \ddots \\ 0 & & & \mathcal{C}_{(X-\lambda_{im_i})^{\mu_{im_i}}} \end{pmatrix}$$

Nous allons maintenant montrer que pour tout λ et n, le bloc de Jordan $J(\lambda, n)$ est semblable à la matrice compagne $\mathcal{C}_{(X-\lambda)^n}$.

Là encore, nous avons d'abord prouvé ce résultat en donnant explicitement la matrice de passage. Mais la vérification est longue et fait intervenir des calculs avec des coefficients binomiaux. Nous avons donc finalement appliqué la même méthode que précédemment, c'est-à-dire utilisé le théorème `similar_fundamental` pour ramener le problème à une équivalence, puis le lemme `Smith_gcdr_spec` pour déterminer les facteurs invariants des matrices. Nous montrons ainsi que le seul facteur invariant de $J(\lambda, n)$ est le polynôme $(X - \lambda)^n$.

Nous pouvons maintenant établir que la matrice $\mathcal{C}_{p_i}$ est semblable à :

$$\begin{pmatrix} J_{i1} & & 0 \\ & \ddots & \\ & & \ddots \\ 0 & & & J_{im_i} \end{pmatrix}$$

avec $J_{ij} = J(\lambda_{ij}, \mu_{ij})$.

Donc chaque bloc de la forme normale de Frobenius de A, est semblable à une matrice comme celle ci-dessus où les (λ_{ij}, μ_{ij}) sont les paires composées des racines des facteurs invariants et de leur multiplicité. Cela explique la définition de la forme normale de Jordan donnée plus haut, et démontre aussi que la forme normale de Frobenius de A est semblable à la forme normale de Jordan de A.

5.4. Diagonalisation

Nous venons de voir que, dans un corps clos, toute matrice est semblable à sa forme normale de Jordan, qui est triangulaire supérieure. Ceci nous donne directement un théorème de trigonalisation. Nous allons voir maintenant dans quelles conditions une matrice est diagonalisable.

Nous avons vu précédemment que la forme normale de Jordan d'une matrice A est composée de blocs de Jordan $J(\lambda, k)$, où λ est une racine d'un facteur invariant de A, et k sa multiplicité. Or k est aussi la taille du bloc $J(\lambda, k)$. Donc si $k = 1$, la forme normale de Jordan est diagonale. Comme les facteurs invariants se divisent successivement, il suffit que le dernier de la liste soit à racines simples pour que tous les facteurs invariants soient à racines simples. Or il se trouve que le dernier facteur invariant est le polynôme minimal de A. Donc il suffit que le polynôme minimal de la matrice A soit à racines simples pour que A soit diagonalisable :

```
Lemma diagonalization n (A : 'M[R]_n.+1) : uniq (root_seq (mxminpoly A)) ->
  similar A (diag_mx_seq n.+1 n.+1 (root_seq (char_poly A))).
```

où `uniq` est un prédicat exprimant que la séquence donnée en argument est sans doublons.

Conclusion

Dans ce travail, nous avons choisi de déduire l'existence des formes de Frobenius et de Jordan à partir de la théorie des facteurs invariants d'une matrice sur un anneau principal. Ce point de vue est naturel, mais n'est pas toujours adopté. De nombreuses présentations utilisent par exemple une théorie des endomorphismes cycliques développée pour la circonstance.

Le théorème d'existence de la forme normale de Frobenius a une portée importante car il permet de capturer la structure des endomorphismes d'un espace vectoriel de dimension finie sur un corps quelconque (discret dans notre contexte). Le travail présenté dans cet article constitue à notre connaissance la première formalisation de ce résultat dans un assistant de preuve.

Un autre intérêt de notre développement est de présenter des définitions et des propriétés des matrices diagonales par blocs ou encore des matrices compagnes, qui pourront être réutilisées car ce sont des notions de base en algèbre matricielle. De la même manière, bien que notre développement traite de notions mathématiques relativement élémentaires, il présente la particularité de reposer sur plusieurs formalisations préexistantes, qu'il contribue à éprouver. Citons par exemple l'existence de la forme normale de Smith [CDMS12], les corps algébriquement clos [Coh12] ou encore la formule de Binet-Cauchy et les définitions de sous-matrice et de mineur associées [Sil12].

Dans la plupart des cas, ces travaux antérieurs se sont révélés facilement adaptables à notre contexte. La preuve du théorème fondamental de similitude sur un corps notamment, exposée section 2, a bénéficié de la grande modularité de la théorie de la division de polynômes fournie par SSReflect.

En revanche, comme indiqué dans la section 1, nous pourrions utiliser une définition plus naturelle de matrice diagonale par blocs si une structure intermédiaire d'anneau n'excluant pas l'anneau trivial était fournie par SSReflect. Mais l'ajout d'une telle structure peut avoir des conséquences qui restent à étudier, comme l'allongement de la hiérarchie globale des structures algébriques, ce qui pourrait stresser trop fortement l'implémentation actuelle de Coq. Un autre point améliorable, indépendant du

problème précédent, est que notre définition ne peut être itérée. En effet, il n'est pas possible en l'état de définir une matrice diagonale par blocs dont les blocs sont eux-mêmes des matrices diagonales par blocs, l'obstacle technique étant le même que celui décrit à la section 4 pour les matrices compagnes.

Notre prochain objectif est d'utiliser les notions définies ici comme spécification pour une implémentation efficace d'un algorithme de calcul de la forme de Frobenius. Par exemple, celui décrit dans [Sto01] a une complexité en $\mathcal{O}(n^\omega \log n \log\log n)$ pour une implémentation du produit matriciel en $\mathcal{O}(n^\omega)$ et serait un bon candidat car il est déterministe. Une sous-routine de cet algorithme est connue sous le nom d'algorithme de Keller-Gehrig [KG85] et a son intérêt propre car elle permet de calculer le polynôme caractéristique d'une matrice de taille n en $\mathcal{O}(n^\omega \log n)$.

Pour implémenter et prouver la correction de ces algorithmes, nous prévoyons d'utiliser la méthodologie basée sur des raffinements ainsi que l'implémentation certifiée du produit de matrices de Strassen décrites dans [DMS12].

Enfin, le lecteur ou la lectrice intéressé(e) trouvera les sources complètes du développement formel présenté dans cet article, ainsi qu'une documentation hypertexte, à l'adresse : `http://www-sop.inria.fr/members/Maxime.Denes/canonical_forms`.

Remerciements

Nous tenons à remercier Cyril Cohen pour les échanges constructifs que nous avons eus autour de ce travail, et notamment son aide dans l'utilisation de la théorie des corps algébriquement clos malgré son statut expérimental dans SSReflect. Nous remercions également Laurence Rideau et Yves Bertot pour leurs relectures attentives et leur conseils qui ont contribué à améliorer la présentation de cet article.

Références

[CDMS12] C. Cohen, M. Dénès, A. Mörtberg et V. Siles : Smith Normal form and executable rank for matrices, 2012. `http://wiki.portal.chalmers.se/cse/pmwiki.php/ForMath/ProofExamples`.

[Coh12] C. Cohen : *Formalized algebraic numbers : construction and first order theory.* Thèse de doctorat, École Polytechnique, 2012.

[Coq12] Coq : The Coq Proof Assistant Reference Manual, version 8.4. Rapport technique, 2012.

[CSLY04] K. Chemla, G. Shuchun, G. E. R. Lloyd et T. Yasumoto : *Les neuf chapitres le classique mathématique de la Chine ancienne et ses commentaires.* Dunod, Paris, 2004.

[DMS12] M. Dénès, A. Mörtberg et V. Siles : A refinement-based approach to computational algebra in coq. *In Interactive Theorem Proving*, volume 7406 de *LNCS*, pages 83–98, 2012.

[Gie95] M. Giesbrecht : Nearly optimal algorithms for canonical matrix forms. *SIAM Journal on Computing*, 24(5):948–969, octobre 1995.

[GM09] G. Gonthier et A. Mahboubi : A Small Scale Reflection Extension for the Coq system. Rapport technique, Microsoft Research INRIA, 2009. `http://hal.inria.fr/inria-00258384`.

[KG85] W. Keller-Gehrig : Fast algorithms for the characteristics polynomial. *Theoretical Computer Science*, 36:309–317, janvier 1985.

[OB08] S. Ould Biha : Formalisation des mathématiques : une preuve du théorème de Cayley-Hamilton. *In JFLA (Journées Francophones des Langages Applicatifs)*, pages 1–14, Etretat, France, 2008.

[Sil12] V. Siles : A formal proof of the Cauchy-Binet formula, 2012. `http://wiki.portal.chalmers.se/cse/pmwiki.php/ForMath/ProofExamples`.

[Sto01] A. Storjohann : Deterministic computation of the frobenius form. *In 42nd IEEE Symposium on Foundations of Computer Science*, pages 368–377, 2001.

[Wed08] J.H.M. Wedderburn : *Lectures on Matrices*. Colloquium Publications. American Mathematical Society, 2008.